kindred

Nolla Utanförskapet

Intro to database

Part 2

© Kindred Group plc 2020

Agenda

- What is a KEY
- Constraints
- Database normalization
- ↔ Database types of relations
- ACID properties
- JOINs
- Let’s start create the DB for the project

© Kindred Group plc 2022

Classified as General

What is a KEY

© Kindred Group plc 2022

Classified as General

🔑 What is a KEY

In SQL, a KEY is a value used to identify a record in table, uniquely. It can be defined by a single column or by a combination of multiple columns.

🔑 Primary Key

🗝 Foreign Key

© Kindred Group plc 2022

Classified as General

Primary Key

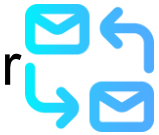
1. A primary key cannot be NULL
2. A primary key value must be unique
3. The primary key values should rarely be changed

© Kindred Group plc 2022

Classified as General

🗝 □ Foreign Key

Foreign Key references the primary key of another Table. It helps connect our Tables (form relations).

1. A foreign key can have a different name from its primary key
2. It ensures rows in one table have corresponding rows in another 
3. Unlike the Primary key, they do not have to be unique (most often they aren't)
4. Foreign keys can be null even though primary keys can not

© Kindred Group plc 2022

Classified as General

🗝 □ Foreign Key

forms of address

ID	Form of address
1	Mrs.
2	Ms.
3	Mr.

clients

ID	First Name	Last Name	Form of address
1	John	Smith	3
2	Jane	Philips	2
3	Mary	Carol	2
4	Michael	Bay	
5	**Amelie**	**Allouette**	4

© Kindred Group plc 2022

Classified as General

Constraints

© Kindred Group plc 2022

Classified as General

Constraints

Constraints are the rules enforced on data columns on a table and are used to prevent invalid data from being entered into the database.

This ensures the **accuracy** and **reliability** of the data in the database.

NOT NULL	PRIMARY KEY
UNIQUE	FOREIGN KEY
	CHECK

© Kindred Group plc 2022

Classified as General

Constraints

`NOT NULL` - Ensures that a column cannot have NULL value

```
CREATE TABLE student (
    id INT PRIMARY KEY,
    name TEXT NOT NULL,
    age INT NOT NULL,
    address CHAR(50)
);
```

© Kindred Group plc 2022

Classified as General

Constraints

`UNIQUE` - Ensures that all values in a column are different

```
CREATE TABLE student (
    id INT PRIMARY KEY,
    name TEXT NOT NULL,
    age INT NOT NULL,
    nickname TEXT NOT NULL UNIQUE,
    address CHAR(50)
);
```

© Kindred Group plc 2022

Classified as General

Constraints

`PRIMARY KEY` - Uniquely identifies each row/record in a database table

```
CREATE TABLE student (
    id INT PRIMARY KEY,
    name TEXT NOT NULL,
    age INT NOT NULL,
    address CHAR(50)
);
```

`PRIMARY KEY` is `NOT NULL` and `UNIQUE` by *default*

© Kindred Group plc 2022

Classified as General

Constraints

`FOREIGN KEY` – Constrains* data based on columns in other tables

```
CREATE TABLE mentor (
    id INT PRIMARY KEY,
    name TEXT NOT NULL,
    age INT NOT NULL,
    student_id INT REFERENCES student(id),
    address CHAR(50)
);
```

* makes sure we reference an existent student when adding a new one, but also have a right approach on updates and deletes

© Kindred Group plc 2022

Classified as General

Constraints

`FOREIGN KEY` – on **deleting/updating** we have 3 options:

1. Disallow deleting a referenced student
2. Delete ‘child’ records as well
3. Something else

© Kindred Group plc 2022

Classified as General

Constraints

FOREIGN KEY

1. Disallow deleting a referenced student

```
CREATE TABLE mentor (
    id INT PRIMARY KEY,
    name TEXT NOT NULL,
    age INT NOT NULL,
    student_id INT REFERENCES student(id) ON DELETE RESTRICT,
    address CHAR(50)
);
```

© Kindred Group plc 2022

Classified as General

Constraints

FOREIGN KEY

2. Delete ‘child’ records as well

```
CREATE TABLE mentor (
    id INT PRIMARY KEY,
    name TEXT NOT NULL,
    age INT NOT NULL,
    student_id INT REFERENCES student(id) ON DELETE CASCADE,
    address CHAR(50)
);
```

© Kindred Group plc 2022

Classified as General

Constraints

FOREIGN KEY

3. Something else (SET NULL /SET DEFAULT)

```
CREATE TABLE mentor (
    id INT PRIMARY KEY,
    name TEXT NOT NULL,
    age INT NOT NULL,
    student_id INT REFERENCES student(id) ON DELETE SET NULL,
 -- student_id INT REFERENCES student(id) ON DELETE SET DEFAULT,
    address CHAR(50)
);
```

© Kindred Group plc 2022

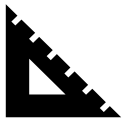
Classified as General

Constraints

CHECK - Ensures that all values in a column satisfy certain conditions

```
CREATE TABLE student (
    id INT PRIMARY KEY,
    name TEXT NOT NULL,
    age INT NOT NULL CHECK(age>0),
    address CHAR(50)
);
```

© Kindred Group plc 2022
Classified as General



Database normalization

© Kindred Group plc 2022

Classified as General

Database normalization

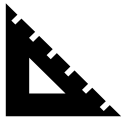
What? How? Why?

Normalization is the process of organizing data in a database.

This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

© Kindred Group plc 2022

Classified as General



Database normalization

Common Types

First normal form (1NF)

Second normal form (2NF)

Third normal form (3NF)

© Kindred Group plc 2022

Classified as General

Database normalization

First normal form (1NF) **No repeating groups: Tables should only have 2**

Student	Age	Classes
Helen	24	Mathematics, English
Joe	25	History
Mary	26	Biology, Geograhy, English

OR

Student	Age	Geography	Mathematics	English	History	Biology
Helen	24	0	1	1	0	0
Joe	25	0	0	0	1	0
Mary	26	1	0	1	0	1

Student	Age	Class
Helen	24	Mathematics
Helen	24	English
Joe	25	History
Mary	26	Biology
Mary	26	Geography
Mary	26	English

© Kindred Group plc 2022

Classified as General

Database normalization

Second normal form (2NF)

1NF + all attributes within the entity should depend solely on the unique identifier of the entity

purchases

Customer Id	Storel d	Location of purchase
2	1	Sweden
2	2	France
44	1	Sweden
1	3	Belgium
6	2	France

2

1

stores

ID	Location
1	Sweden
2	France
3	Belgium

purchases

Customer Id	Storel d	Lo c hase
2	1	we n
2	2	ranc
44	1	veden
1	3	B ium
6	2	France

© Kindred Group plc 2022

Classified as General

Database normalization

Third normal form (3NF)

2NF + eliminate (move it outside into another table) all fields that do not depend on the key

clients

ID	First Name	Last Name	Form of address
1	John	Smith	Mr.
2	Jane	Philips	Ms.
3	Amelie	Allouette	Mrs.

forms_of_address

ID	Form of address
1	Mrs.
2	Ms.
3	Mr.

clients

ID	First Name	Last Name	Form of address
1	John	Smith	3
2	Jane	Philips	2
3	Amelie	Allouette	1

© Kindred Group plc 2022

Classified as General

↔□ Database types of relations

© Kindred Group plc 2022

Classified as General

↔□ Database types of relations

Types

One-To-One

One-To-Many

Many-To-Many

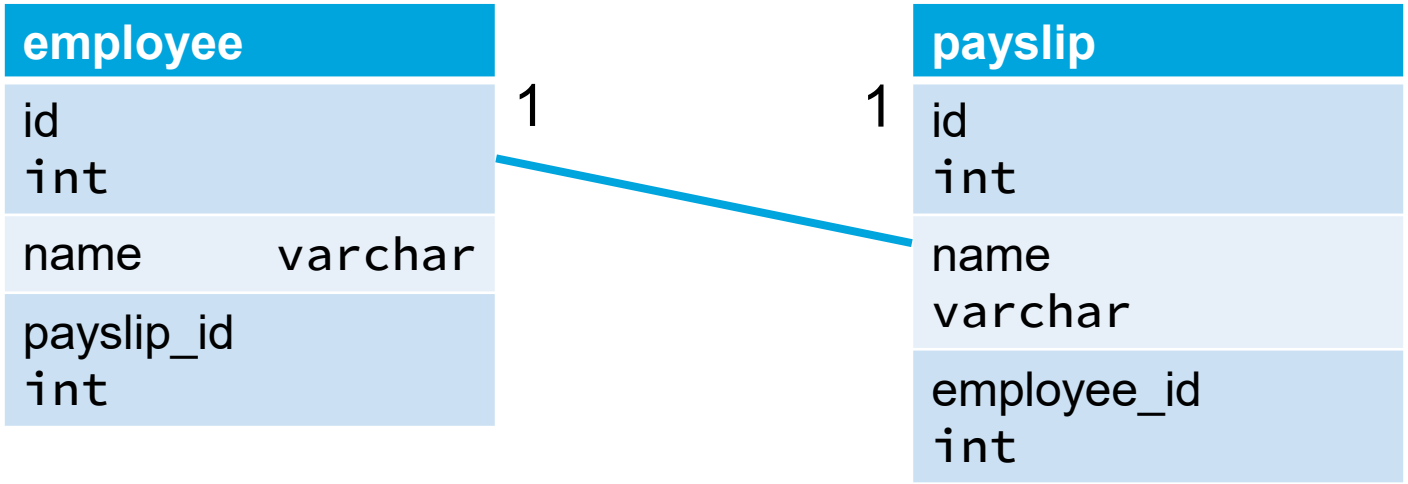
© Kindred Group plc 2022

Classified as General

↔□ Database types of relations

One-To-One Relation

A row in table A can have only one matching row in table B, and vice versa.



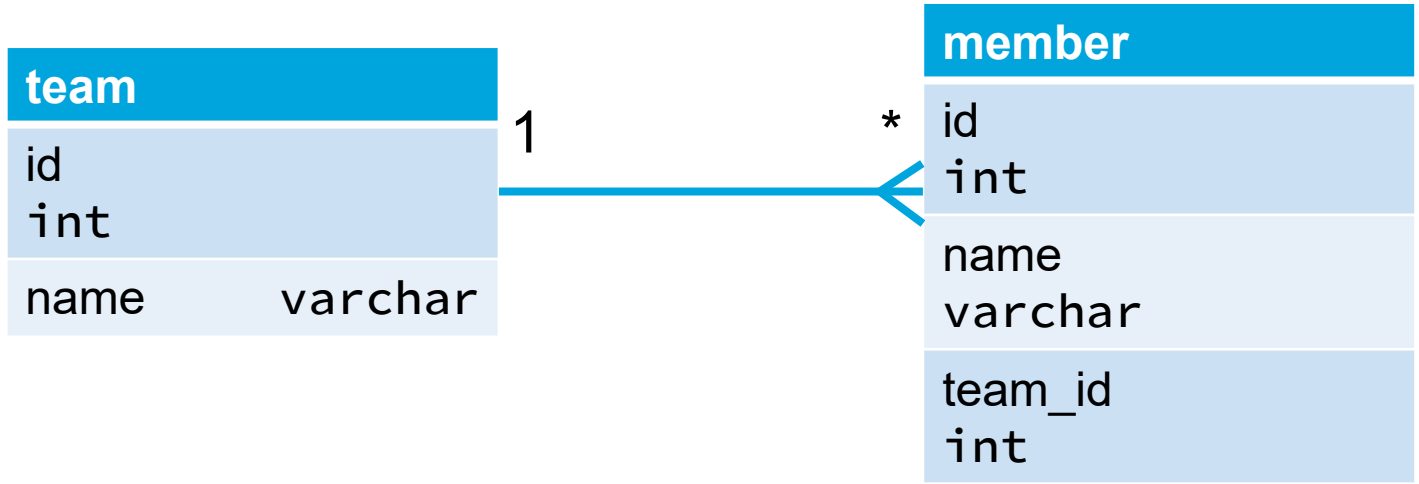
© Kindred Group plc 2022

Classified as General

↔□ Database types of relations

One-To-Many/Many-To-One Relation

A row in table A can have many matching (or none) rows in table B, but a row in table B can have only one matching row in table A.



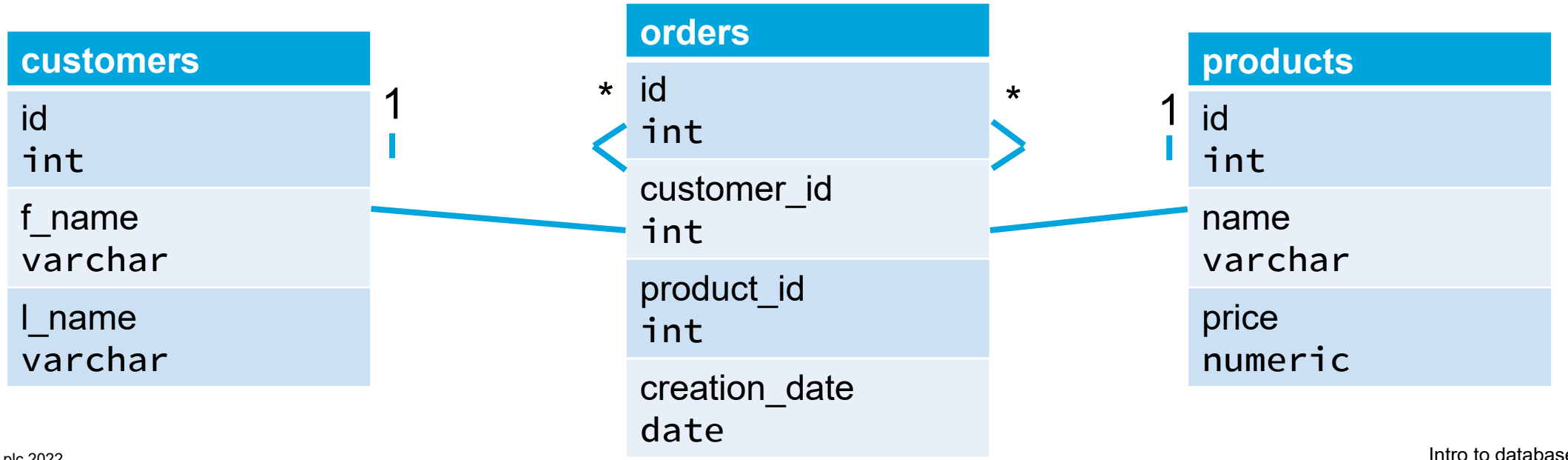
© Kindred Group plc 2022

Classified as General

↔□ Database types of relations

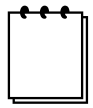
Many-To-Many Relation

A row in table A can have many matching rows in table B, and vice versa.



© Kindred Group plc 2022

Classified as General



ACID properties

© Kindred Group plc 2022

Classified as General

🗒 ACID properties

A transaction is a *set of read/write operations* organized to be run successfully as a *whole*.

Example:

An e-wallet payment transaction:

1. Check if current balance is sufficient (read)
2. Update current balance by subtracting the amount (write)
3. Other required action that might be needed like:
   a. Updating a transaction history (write)
   b. Check if customer is allowed to perform such payment (read)

A - Atomicity
C - Consistency
I - Isolation
D - Durability

© Kindred Group plc 2022
Classified as General

🗒️ ACID properties

Atomicity takes individual operations that are a part of a transaction and refers to them into an **all-or-nothing unit of work**, making the transaction a successful one only if *all contained operations succeed*.

© Kindred Group plc 2022

Classified as General

🗒️ ACID properties

Consistency ensures that a transaction can only bring the database from one valid state to another.

(any data written to the database must be valid according to all defined rules within the database)

© Kindred Group plc 2022

Classified as General

🗒️ ACID properties

Isolation hides uncommitted state changes from the outside world.

(the effects of an incomplete transaction might not even be visible to other transactions)

© Kindred Group plc 2022

Classified as General

🗒️ ACID properties

Durability guarantees that once a transaction has been committed, it will remain committed even in the case of a system failure (e.g., power outage or crash).

© Kindred Group plc 2022

Classified as General

JOINs

© Kindred Group plc 2022

Classified as General

🔗 JOINs

The PostgreSQL database supports mainly five types of Joins

Inner/ Simple Join

Left Outer Join

Right Outer Join

Full Outer Join

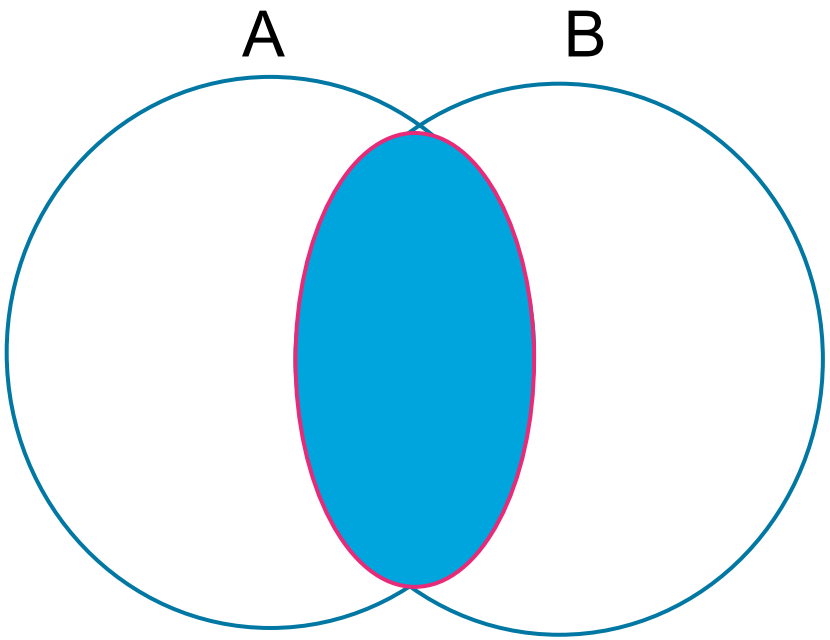
Cross Join

© Kindred Group plc 2022

Classified as General

JOINs

Inner Join



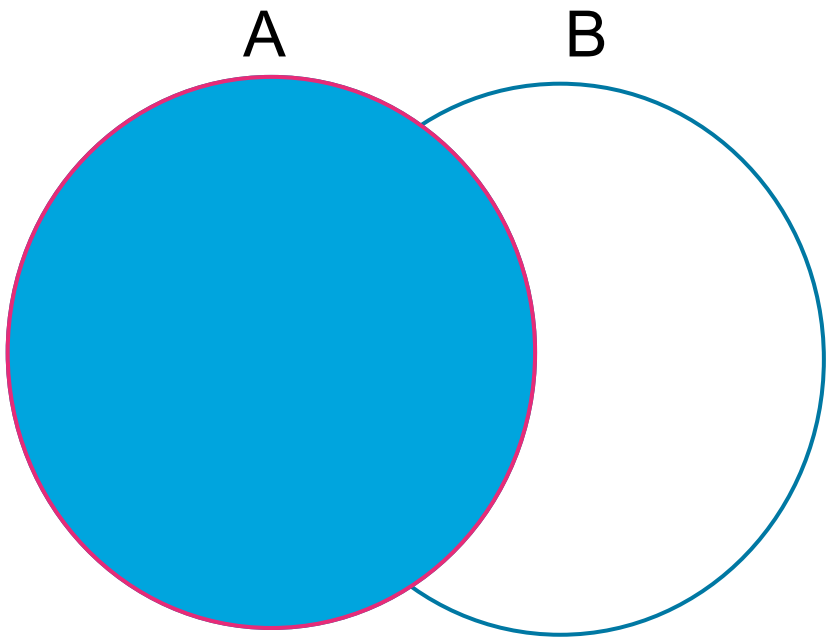

```
SELECT A.column,
       B.column
FROM A INNER JOIN B ON
A.some_column = B.some_column;
```

© Kindred Group plc 2022

Classified as General

JOINs

Left Outer Join

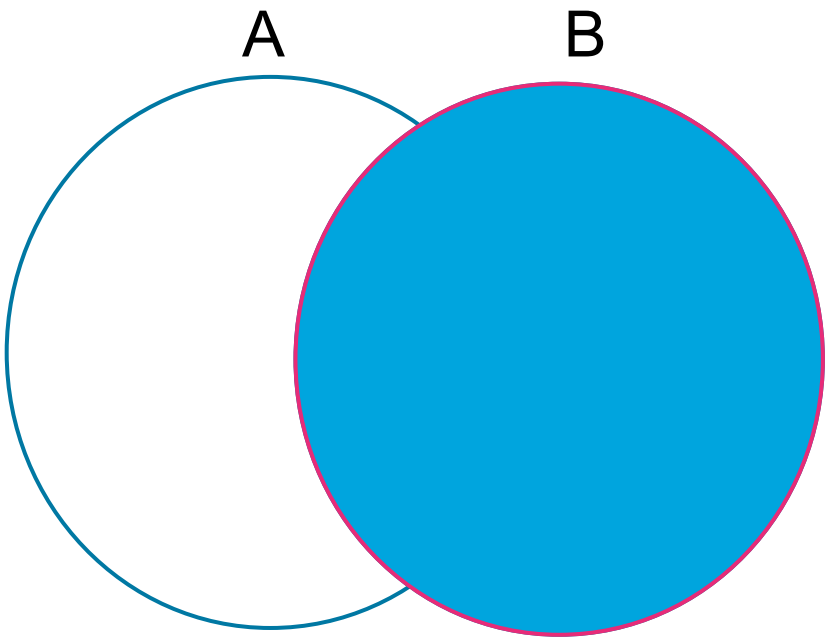



```
SELECT A.column,
       B.column
FROM A LEFT JOIN B ON
A.some_column = B.some_column;
```

© Kindred Group plc 2022
Classified as General

JOINs

Right Outer Join



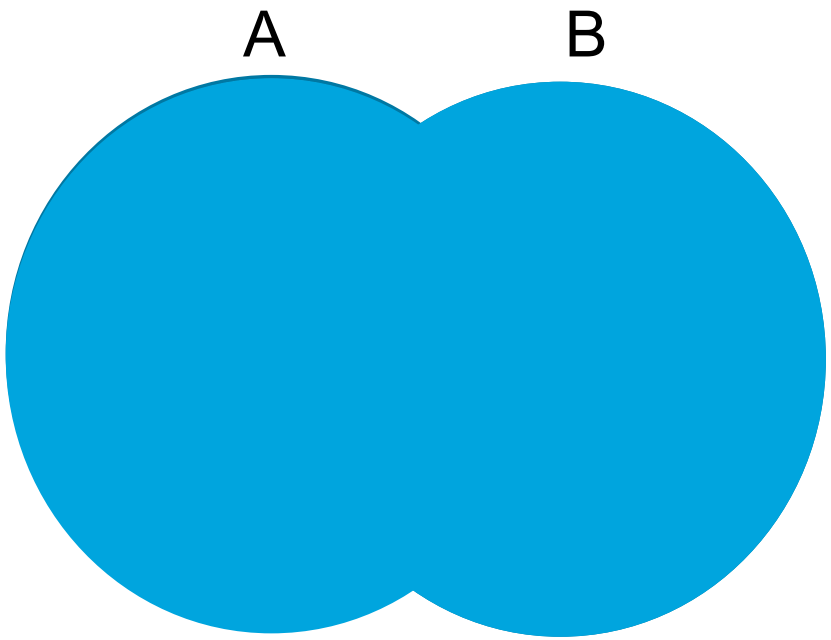

```
SELECT A.column,
       B.column
FROM A RIGHT JOIN B ON
A.some_column = B.some_column;
```

© Kindred Group plc 2022

Classified as General

JOINs

Full Outer Join



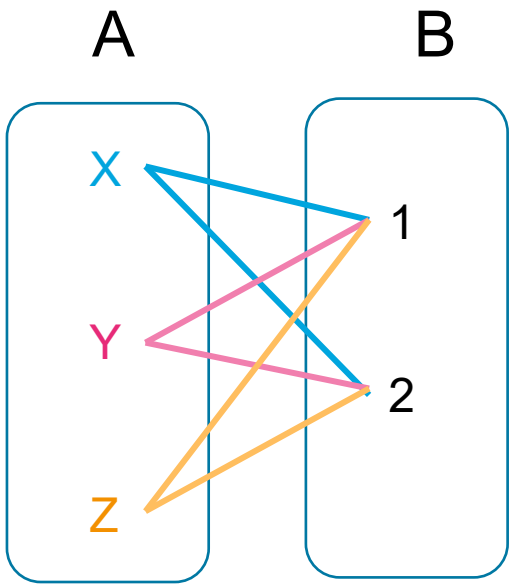

```
SELECT A.column,
       B.column
FROM A FULL OUTER JOIN B ON
A.some_column = B.some_column;
```

© Kindred Group plc 2022

Classified as General

JOINs

Cross Join (cartesian product)

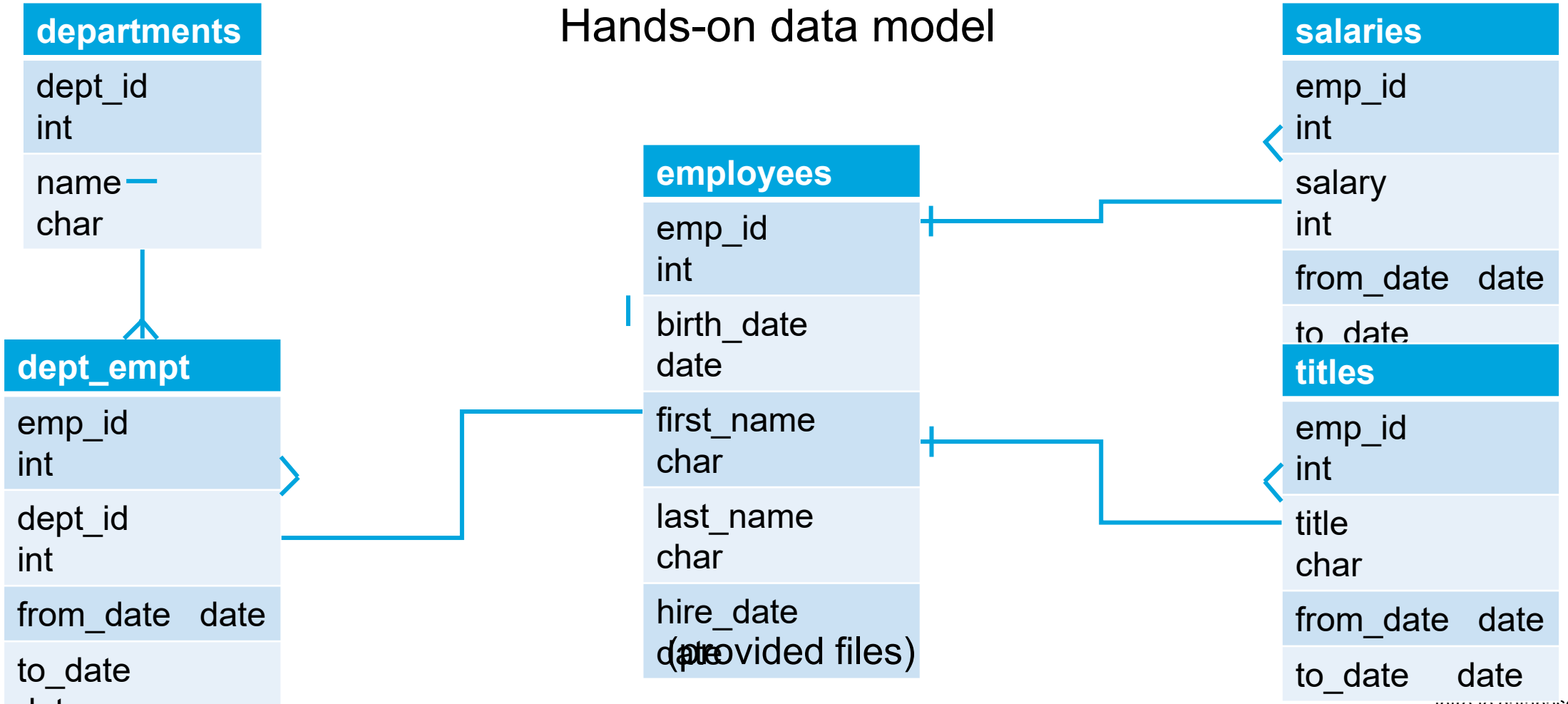



```
SELECT A.column,
       B.column
FROM A CROSS JOIN B ON
A.some_column = B.some_column;
```

© Kindred Group plc 2022

Classified as General

JOINs



© Kindre

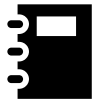
Classified as General

🔗 JOINs

1. Get only the employees having ‘engineer’ in title.
2. Get only the employees from Payments department.
3. Get all the employees and their current department.

© Kindred Group plc 2022

Classified as General



Let’s start create the DB for the project

© Kindred Group plc 2022

Classified as General

© Kindred Group plc 2020

Classified as General